Project1 CSE3027 Spring 2023

Division of Computer Science - 2021051258 Choi Minjun (컴퓨터학부 - 최민준)

1. Server design

Server design can be divided into six major steps. These steps include socket creation, address binding, listening for connections, accepting client connections, and transmitting data. In order to continuously receive requests from clients, the server program loops endlessly from the accept phase to the client socket's close phase.

Before actually opening the server, it receives arguments as follows. If no arguments are provided, an error is handled as a port number has not been provided.

```
project1_2021051258_Choi_Minjun$ ./server 8080
```

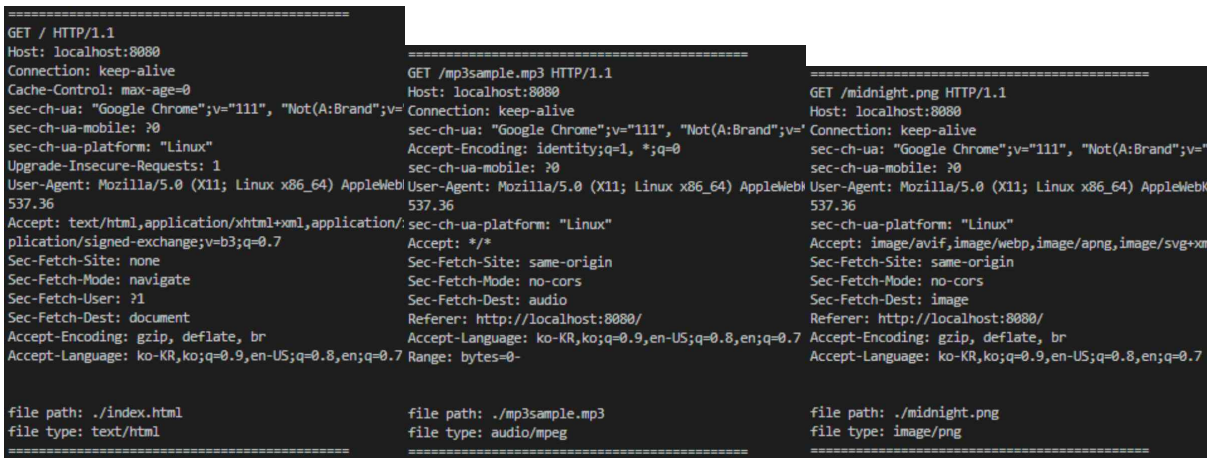
If the port number is successfully received, a socket is created for IPv4 and TCP environments. Then, an IP address and port number are assigned to the socket (binding). The setsockopt() function is used to set the socket's option to turn off the server immediately upon program termination. If the address binding is completed successfully, the server waits for a client connection. Once the connection is established successfully, the server connects to the client and reads the client's request. Based on the read value, the server parses what the client wants and opens the requested file to return its content. After completing these steps, the server closes the file and the client socket. Then, the server accepts the client socket again and repeats the above steps until the user forcibly terminates the program.

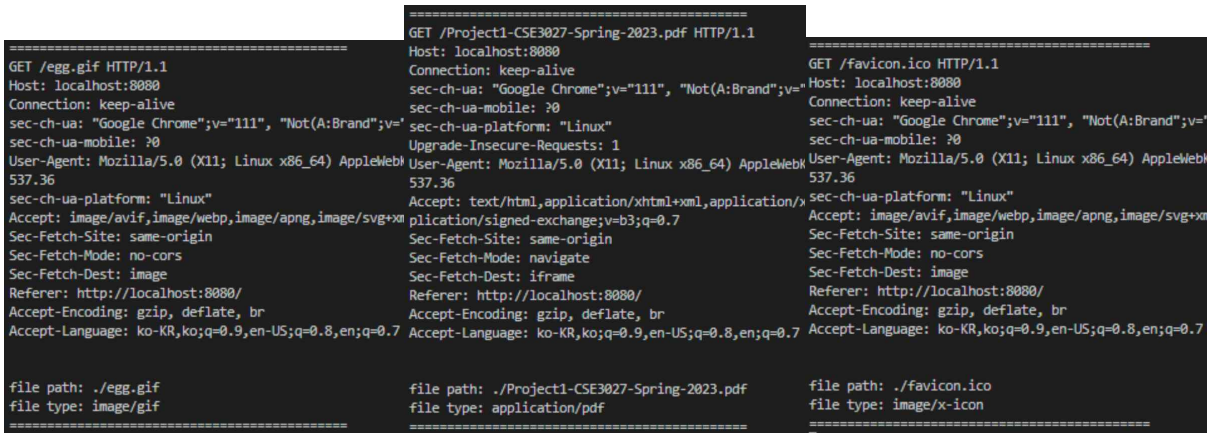
2. Difficults

The most challenging part for me was understanding the given example before starting the assignment. I did not have a complete understanding of the detailed steps to open a server, so I invested a lot of time in understanding the example. However, the example helped me understand the process more easily. Additionally, providing what the client requested was also difficult, especially the process of parsing the request which was more challenging than I had initially thought.

3. Sample Output

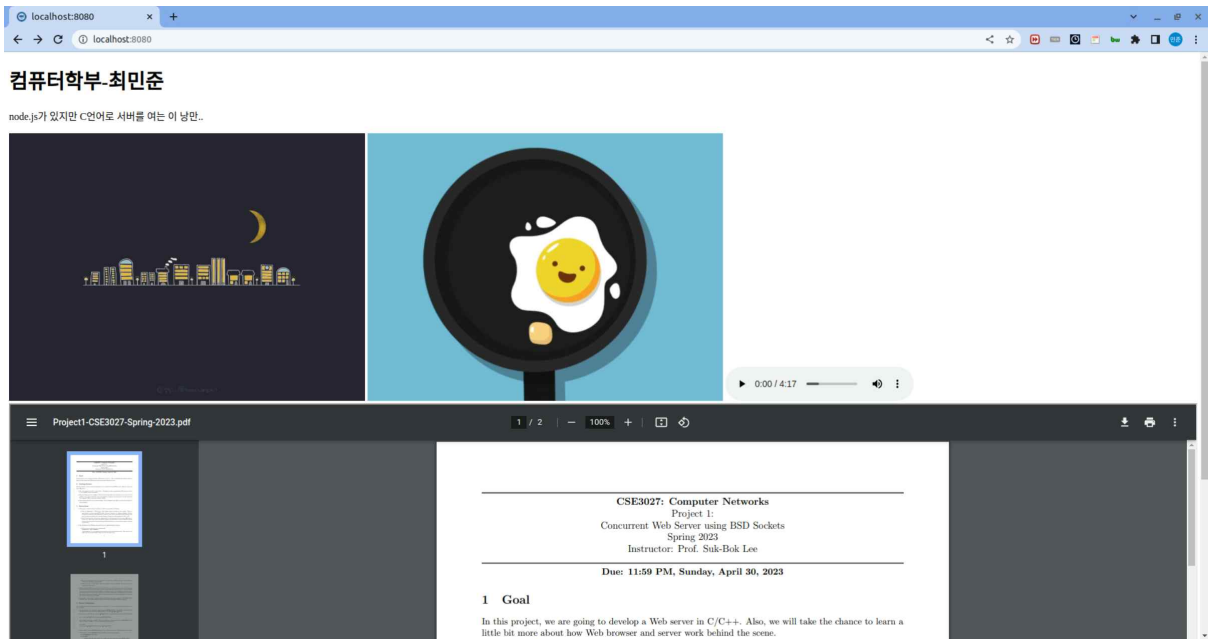
The following is the Sample Output:

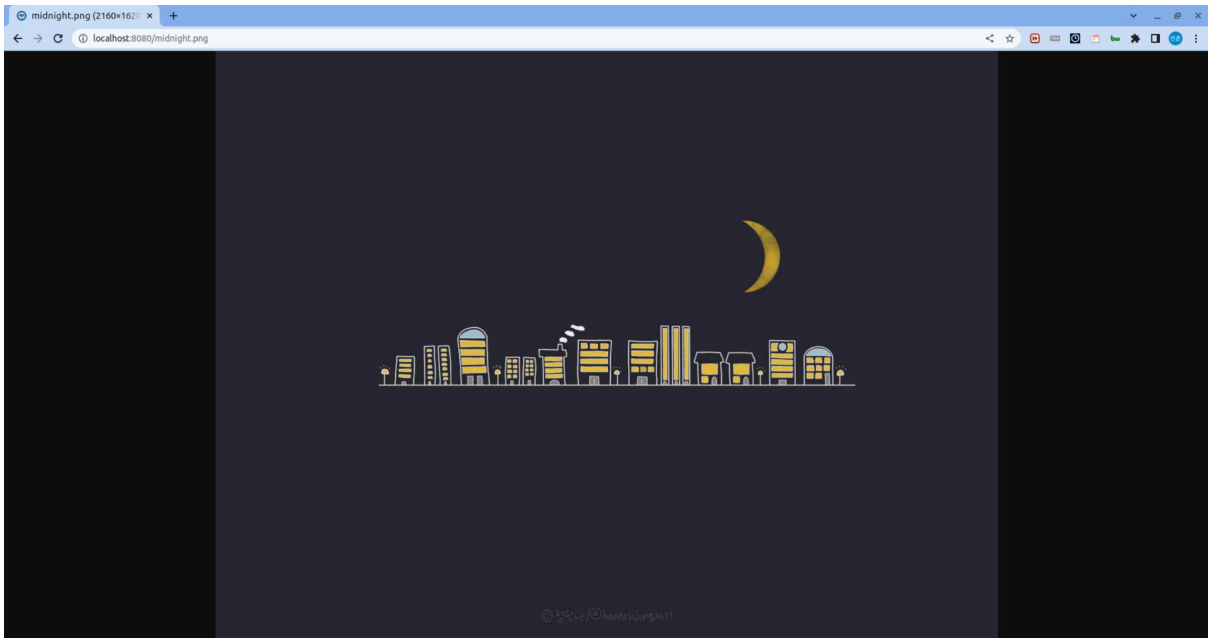




위의 총 6개의 출력은 맨 처음 주소창에 localhost:8080을 입력했을 때 출력되는 것들이다. 코드에서는 사용자가 만약 루트에 접근하였다면 index.html파일로 접근하도록 설정하였다. 따라서 루트로 접근하면 index.html의 내용들이 클라이언트에 보이게 된다. 이때 index.html파일의 내부에는 png, gif, mp3, pdf 들이 들어있기 때문에 위와 같은 요청이 발생한 것이다. html요청에서 favicon.ico에 대한 요청도 존재했기에 이에 대한 요청도 추가적으로 처리해주었다. 아래는 루트에 대한 접근, midnight.png에 대한 접근이다.

The six outputs shown above are what would be displayed when entering localhost:8080 in the address bar. The code is set up so that if the user accesses the root, they will be directed to the index.html file. Therefore, accessing the root will display the contents of the index.html file to the client. Since the index.html file contains png, gif, mp3, and pdf files, these requests were generated as shown above. There was also a request for favicon.ico in the html request, so I also handled this request separately. Below are examples of accessing “localhost:8080/” and “localhost:8080/midnight.png”.




midnight.png (2160×162
localhost:8080/midnight.png